Histogram of Body Poses and Spectral Regression Discriminant Analysis for Human Action Categorization

Conference Paper · January 2010

2 authors, including:

Xiuli Chen
University of Birmingham
36 PUBLICATIONS   **352** CITATIONS

Histogram of Body Poses and Spectral Regression Discriminant Analysis for Human Action Categorization

Ling Shao
ling.shao@sheffield.ac.uk

Xiuli Chen
elp09xc@sheffield.ac.uk

Department of Electronic and Electrical Engineering

The University of Sheffield, UK

Abstract

This paper explores a recently proposed and rarely reported subspace learning method, Spectral Regression Discriminant Analysis (SRDA) [1, 2], on silhouette based human action recognition. The recognition algorithm adopts the Bag of Words (BoW) model combined with the action representation based on Histogram of Body Poses sampled from silhouettes in the video sequence. In addition, we compare the performance of SRDA for dimensionality reduction with several traditional subspace learning methods, such as Principle Component Analysis (PCA), supervised Locality Preserving Projections (LPP), unsupervised LPP and Neighbourhood Preserving Embedding (NPE). Experimental results show that Histogram of Human Poses combined with SRDA or its kernel version, SRKDA, can achieve 100% recognition accuracy for the Weizmann human action dataset, which is better than any published results on the same dataset.

1. Introduction

The human action recognition technology can be applied on a wide range of fields, such as video surveillance for security-sensitive areas, content based video search and automatic interpretation of video sequences for hi-tech digital libraries, man-machine interaction and analysis of sports events, etc. These interesting and promising applications motivate more and more researchers from the fields of computer vision, pattern recognition and machine learning to focus on this popular topic.

Many approaches for human action recognition have been developed. There are mainly two groups of methods: part-based and holistic. Part-based, also called interest point based, methods [3, 4, 5] are usually more resilient to background clutter and occlusion. However, the global and dynamic information of the human performing the action is neglected during extraction of the sparse feature points. Holistic methods have more global information to rely on and can perform very well, if the foreground subjects are reliably segmented from the background, which is often a difficult task, especially when the background is complex and changing. However, foreground segmentation is not the focus of this paper and we assume reliable silhouettes of the human body are available. In silhouette based human action recognition, dimensionality reduction techniques are adopted to project the original

data which are usually in a very high dimensional space to a low dimensional one. The most popular reduction methods are Principal Component Analysis (PCA) [6] and Linear Discriminant Analysis (LDA) [7]. Another popular linear algorithm is Locality Preserving Projections (LPP) [8]. However, these linear dimensionality reduction algorithms fail on nonlinear problems which are easy to encounter in real world. Consequently, nonlinear methods are proposed to deal with the nonlinear problems, such as Isometric Feature Mapping (Isomap) [9], Locally Linear Embedding (LLE) [10], Laplacian Eigenmaps (LE) [11] and Diffusion Maps [12]. In addition, there are kernel-based methods, which try to map the nonlinear data to a feature space using a nonlinear mapping function, while the tricky part is that they simply compute the inner product on the feature space using a kernel function instead of knowing the nonlinear mapping function explicitly [13]. Wang and Suter [14] explored the intrinsic structure of the articulated action space using Kernel Principle Component Analysis (KPCA), and employ Factorial Conditional Random Field (FCRF) for activity modelling and recognition. Kernel Fisher Discriminant Analysis (KFDA) proposed in [15] is the kernel-based nonlinear extension of LDA. In [16], Baudat and Anouar presented Generalized Discriminant Analysis (GDA) to address nonlinear discriminant analysis using kernel function operator.

However, there exists a common problem among the above algorithms, i.e. the large amount of computation caused by the eigen-decomposition of dense matrices. It is expensive in both time and memory. In order to solve this problem, Cai *et al*. [1, 2] proposed a new subspace learning algorithm, Spectral Regression Discriminant Analysis (SRDA). Instead of doing the eigen-decomposition of dense matrices, SRDA simply casts projective function learning into a regression framework, which results in linear-time complexity, compared with cubic-time complexity in the other subspace learning algorithms. SRDA has shown to be a promising approach in several applications, such as face recognition [2], speech letter recognition [17], text clustering and categorization [1], and handwritten digital classification [17]. Moreover, there is an extended version of SRDA, Spectral Regression Kernel Discriminant Analysis (SRKDA) [2], for nonlinear cases.

In this paper, we aim to investigate SRDA and SRKDA for dimensionality reduction in silhouette based human action recognition and compare its performance with other subspace learning methods. The main contribution of our work is listed below:

(1) We apply Spectral Regression for dimensionality reduction, including SRDA and SRKDA, on silhouette based human action recognition and compare the performance with that of other subspace learning methods, such as PCA, LPP, and NPE.

(2) Histogram of Body Poses (HBP) is proposed to describe a human action sequence. Only raw data sampled from the silhouettes associated with the video sequences are used to represent human poses.

Based on our experimental results, SRDA and SRKDA are proved to outperform other subspace algorithms in terms of dimensionality reduction for human action recognition. The integration with the HBP descriptor achieves better recognition accuracy than any other existing method.

In the following, we first briefly introduce Spectral Regression Discriminant Analysis in Section 2. In Section 3, we describe the Histogram of Body Poses descriptor and the recognition framework. The experimental results and evaluation are presented in Section 4. Finally, we draw conclusion in Section 5.

2. Spectral Regression Discriminant Analysis

In this section, Spectral Regression Discriminant Analysis is outlined. Suppose we have a data set $\{x_i\}_{i=1}^{m} \subset R^n$, X= [$x_1, x_2 \cdots, x_m$], the purpose of the dimensionality reduction algorithms is to find $\{z_i\}_{i=1}^{m} \subset R^d$ which is the low dimensional version of x_i, i.e. d<<n. Generally, it can be nicely interpreted in a *graph embedding* framework, although the motivations can be different. In the *graph embedding* framework, a graph G and an m×m symmetric matrix W are generated to characterize certain statistical or geometric properties of the data set. The graph G has m vertices which represent m data points respectively, while each element of W, $W_{i,j}$, measures the similarity between vertices i and j. Graph embedding aims to find a representation of each vertex using a low dimensional vector which still contains the similarity between the vertex pairs.

Assume that $Y = [y_1, y_2 \cdots y_m]^T$ is the one-dimensional projection of $X = [x_1, x_2 \cdots x_m]$, then we can derive the optimal Y by minimizing

$$\sum_{i,j}(y_i - y_j)^2 W_{i,j} \tag{1}$$

which can be greatly minimized, if how 'close' between y_i and y_j is strictly dependent on how 'close' between x_i and x_j. In other words, the case that the mapped points y_i and y_j, which belong to two neighboring vertices i and j, are far apart will bring serious penalty of this minimization [18]. After some deduction with the definition of the *graph Laplacian L*=D-W

(D is a diagonal matrix: $D_{i,i} = \sum_j W_{j,i}$ or $D_{i,i} = \sum_i W_{i,j}$) [19], we have

$$\sum_{i,j}(y_i - y_j)^2 W_{i,j} = 2y^T Ly \tag{2}$$

Based on Eq. (2), the minimization problem can be simplified to find

$$y^* = \arg\min_{y^T Dy=1} y^T Ly = \arg\min \frac{y^T Ly}{y^T Dy} \tag{3}$$

which can be converted to an equivalent variation as follows:

$$y^* = \arg\max_{y^T Dy=1} y^T Wy = \arg\max_y \frac{y^T Wy}{y^T Dy} \tag{4}$$

To find the optimal **y**'s, the maximum eigen-value in the eigen-problem below needs to be solved:

$$Wy = \lambda Dy \tag{5}$$

where an arbitrary scaling factor in the embedding is removed by the constraint $y^T Dy = 1$. Different choices of W here will lead to many different manifold learning algorithms, such as ISOAMP [20], Laplacian Eigenmap [21], and Locally Linear Embedding [22].

The linear extension of graph embedding (LGE) will be used in the following to provide the mapping for all samples. Assume that a linear function $y_i = f(x_i) = a^T x_i$ has been chosen, we can find $y = X^T a$ with $X = [x_1 \cdots x_m] \in R^{m \times n}$. Then Eq. (4) can be rewritten as:

$$a^* = \arg\max \frac{y^T Wy}{y^T Dy} = \arg\max \frac{a^T XW X^T a}{a^T XD X^T a} \tag{6}$$

The optimal **a**'s are the eigenvectors associated with the maximum eigenvalue of eigen

problem:

$$XW X^T a = XD X^T a \tag{7}$$

This framework, i.e. LGE, with different choices of affinity matrix W and constraint matrix D can form many popular linear dimensionality reduction algorithms, *e.g.*, Linear Discriminant Analysis and Locality Preserving Projection.

For most subspace learning algorithms, the general solution used to solve the eigen-problem in Eq. (7) involves the eigen-decomposition of dense matrices, which results in a huge amount of computational effort. This problem narrows their applicability. In order to address this issue, Cai et al. [1, 2] proposed a new and efficient solution to the Eigen-problem in Eq. (7). As shown in the name, *Spectral Regression* (SR), it combines the spectral analysis of the graph and regression together. The solution is based on the following two theorems.

Theorem 1 *Let y be the eigenvector of the eigen-problem in Eq. (5) with eigenvalue λ. If X^T a = y, then a is the eigenvector of the eigen-problem in Eq. (7) with the same eigenvalue λ.*

Based on Theorem (1), instead of solving the eigen-problem in Eq. (7), they try to find the linear projective functions via two steps:
1. Get **y** by solving the eigen-problem in Eq. (5).
2. Find **a** that satisfies $X^T a = y$. Such **a** might not exist in reality. An alternative way is to find **a**, which can best fit the equation in the least squares sense:

$$a = \arg\min_a \sum_{i=1}^{m} (a^T X_i - y_i)^2 \tag{8}$$

However, the minimization problem (8) is ill posed, when the number of samples is smaller than that of features. The common solution is using regularization method, i.e. imposing a penalty on the norm of **a**:

$$a = \arg\min_a (\sum_{i=1}^{m} (a^T X_i - y_i)^2 + \alpha \|a\|^2) \tag{9}$$

where the parameter $\alpha \geq 0$ is used to control the amount of shrinkage. The regularized least squares is also called ridge regression [23].
The matrix forming the regularized least squares in Eq. (9) is as follows:

$$a = \arg\min_a ((X^T a - y)^T (X^T a - y)) + \alpha a^T a) \tag{10}$$

Derive the right side with respect to **a** vanish, we have

$$(X X^T + \alpha I)a = Xy$$

$$\Rightarrow a = (X X^T + \alpha I)^{-1} Xy \tag{11}$$

When $\alpha > 0$, this regularized solution will not satisfy the linear equation system $X^T a = y$ and **a** will not be the eigenvector of the eigen-problem in Eq. (4). What we are interested is to see when Eq. (11) provides the exact solutions of the eigen-problem in Eq. (4).

Theorem 2 *Suppose y is the eigenvector of the eigen-problem in Eq. (2), if y is in the space spanned by row vectors of X, the corresponding projective function a calculated in Eq. (11) will be the eigenvector of the eigen-problem in Eq. (4) as α decreases to zero.*

When the number of features is larger than that of samples, the sample vectors are usually linearly independent, *i.e.*, rank(X) = m. Consequently, in this case, all the projective

functions calculated by Eq. (11) are the eigenvectors of the eigen-problem in Eq. (3) as α decreases to zero.

The above two-step approach essentially performs regression after the spectral analysis of the graph, thus it is called *Spectral Regression* (SR). For details of SRDA, please refer to [1, 2].

3. Methodology

The methodology used for human action recognition in this work is an extension of the Bag of Words (BoW) model to the visual domain, which is called Bag of Features or Bag of Visual Words. Similar to the BoW model in text classification, the first step is to build a visual vocabulary (also called codebook). One simple way is clustering all the feature vectors obtained from all the training samples by k-means clustering. The centre of each cluster is then defined as a codeword, and the size of the visual dictionary is therefore the number of the clusters. Given N vectors $[x_1; x_2; \cdots x_N]$, we perform k-means clustering to form k clusters, then we can find k cluster centers , i.e. codewords $[w_1, w_2 \cdots w_k]$. Each vector in $[x_1; x_2; \cdots x_N]$ can be assigned to its closest codeword. Each action represented by certain number of vectors can be described as the probability distribution of the codewords in the form of histogram.

In interest point based action recognition, each feature vector is a descriptor calculated around a detected interest point in an action sequence. In this paper, the features we use are body poses instead. Body poses, represented as normalized silhouettes for which the dimensionality is reduced using a certain subspace learning algorithm, from each frame in the training action sequences are fed into the BoW model. After k-means clustering, each action sequence can be described as a histogram of the occurrence of certain poses, which we call 'Histogram of Body Poses' (HBP). Examples of HBP are shown in Figure 1. The x-axis of the histogram is the codewords $[w_1, w_2 \cdots w_k]$ (k=40 here), and the y-axis value of w_i is the number of vectors that belong to w_i over the whole action sequence. These histograms are the data used to train the classifier, e.g. Support Vector Machine (SVM) or K Nearest Neighbours (KNN). During the testing stage, each test sequence is also represented as an HBP and fed into the classifier for action classification. The HBP descriptor is not only invariant to the length and periodicity of the action sequence but also robust to the swiftness and the starting posture of the action.

4. Experiments and Results

4.1 Dataset and preprocessing

A widely used dataset, the Weizmann action dataset [24], is employed in our experiments. In this dataset, there are nine different people and each of them performs ten natural actions, including bending (bend), jumping jack (jack), running (run), walking (walk), jumping-forward-on-two-legs (jump), jumping-in-place-on-two-legs (pjump), galloping ideways (side), waving-one-hand (wave1), and waving-two hands(wave2).
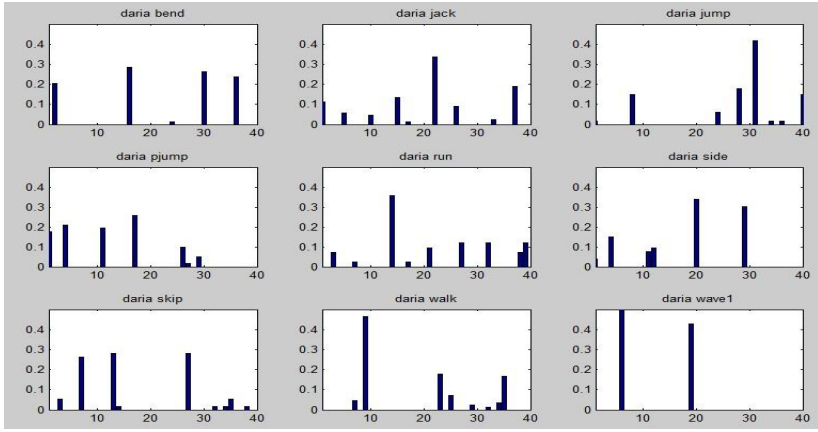
Figure 1: Several actions represented in the form of Histogram of Body Poses (k=40).

There are 93 low-resolution videos (180x144, 25 fps) in total, as some actions are performed twice by certain subjects. When there is a static background, a simple but efficient way to find the silhouettes in a sequence is background subtraction, which is out of the scope of this paper. Therefore, we use the off-the-shelf silhouette masks in [24] directly. In this paper, we employ sequential silhouettes obtained from the original video to describe the dynamic shapes of human actions. In other words, raw data rather than any descriptor calculated on the raw data are used to represent the body pose information in the video frames, which makes our method more efficient. Figure 2 shows the silhouette representation of a running sequence.

Figure 2: Silhouette representation of action 'running'.

In practice, the scale and the position of the subjects often vary in different sequences due to changes in camera distance and viewpoint. Therefore, some pre-processing, including localization of the human body and normalization, should be done to make the silhouette cantered and the aspect ratio equal. Figure 3 shows the steps of the pre-processing.

Figure 3: Pre-processing steps for the human silhouette.

After pre-processing, the silhouette in each frame is cantered and normalized to an M×N image, which is then converted to an M×N dimensional vector consisting of only binary values. Consider that there are c actions represented by c silhouette sequences respectively. For the jth frame of the ith action, it is converted to a vector $V_{i,j}$ by scanning this frame from top-left to bottom-right. N_i is the number of frames of the ith action. The total number of frames is $N_t = N_1 + N_2 + \cdots + N_c$. So the whole data set is represented as:

$$X = [V_{1,1}; V_{1,2}; \cdots V_{1,N_1}; V_{2,1}; V_{2,2}; \cdots V_{2,N_2}; \cdots\cdots; V_{c,1}; V_{c,2}; \cdots V_{c,N_c}]$$
$$= [x_1; x_2; \cdots x_{N_t}]$$

X is a $N_t \times (M*N)$ matrix in which each row represents a frame of in a certain action sequence.

A dimensionality reduction method can be used to embed the data into a lower dimensional subspace, e.g. a 100 dimensional one, for more compact representations. Thus we have a new matrix X^L with size $N_t \times 100$. The new matrix with reduced dimensionality is then used for training the classifier for action recognition.

4.2 Experiment results

Nine-fold cross validation is employed in this paper to evaluate the recognition results of different subspace learning methods, including PCA, Supervised-LPP, unsupervised-LPP, NPE, SR-LDA, and SR-KDA. Specifically, each time, we take actions of one of the nine subjects as test samples and use the action sequences of the remaining eight persons as training data. Both K Nearest Neighbours (KNN) and Support Vector Machine (SVM) are used for classification. For the latter, we adopt the non-linear Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel [25]. The recognition rate is the average of nine runs. Figure 4 shows the confusion matrices of the six subspace learning techniques evaluated. The cross-line values in the tables are the correctly classified rates, while the nonzero values indicate the probabilities of the actions that are mis-classified.

From Figure 4 below, we can observe that:

(1)   'skip' and 'jump' are two relatively more difficult actions for most recognition schemes, e.g. 'skip' is easily mis-classified as 'run', while 'jump' sometimes is confused with 'skip';

(2)   SR-LDA and SR-KDA outperform other subspace learning methods significantly by reaching 100% recognition rate for all actions when using either KNN or SVM as classifier. Even 'skip' and 'jump', which are usually confused in many subspace learning schemes, are correctly classified when SR-LDA and SR-KDA are employed;

(3)   The supervised-LPP performs much better than the unsupervised one.

Table 1: Comparison of different methods.

Methods	Accuracy (KNN)	Accuracy (SVM)	Reduced dimensionality
SR-LDA	100%	100%	9
SR-KDA	100%	100%	9
PCA	93.4%	92.3%	100
LPP	93.4%	93.4%	100
LPP-supervised	98.9%	98.9%	100
NPE	92.3%	91.1%	100

An overall comparison of recognition rates is shown in Table 1. Besides the perfect accuracy performed by SR-LDA and SR-KDA, it can be also noticed that they require a much lower vector dimensionality than the other techniques.

SR-LDA (KNN, k=1)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	1.0	00	00	00	00	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	00	00	00	00	1.0	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

SR-KDA (KNN, k=1)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	1.0	00	00	00	00	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	00	00	00	00	1.0	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

NPE (KNN, k=1)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	.89	00	00	.11	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	.67	00	00	00	.33	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	.11	00	00	.89	00	00	00	00
skip	00	00	.11	00	.11	00	.78	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

PCA (KNN, k=1)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	.78	00	00	00	.22	00	00	00
pjump	.11	00	00	.78	00	.11	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	.11	00	.89	00	00	00	00
skip	00	00	00	00	.11	00	.89	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

LPP-supervised (KNN, k=1)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	1.0	00	00	00	00	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	.11	00	00	00	.89	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

LPP (KNN, k=1)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	.89	00	.11	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	.67	00	00	.11	.22	00	00	00
pjump	.11	00	00	.89	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	.11	00	00	00	.89	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

(a) KNN (k=1)

SR-LDA (SVM)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	1.0	00	00	00	00	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	00	00	00	00	1.0	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

SR-KDA (SVM)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	1.0	00	00	00	00	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	00	00	00	00	1.0	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

NPE (SVM)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	.89	00	00	00	.11	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	.89	00	.11	00	00
skip	00	00	00	00	00	00	1.0	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

PCA (SVM)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	.11	00	.67	00	00	00	.22	00	00	00
pjump	00	00	00	.89	00	.11	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	.22	00	00	00	.78	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	.11	00	00	00	00	00	.89	00
wave2	00	00	00	00	00	00	00	00	00	1.0

LPP-supervised (SVM)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.0	00	00	00	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	1.0	00	00	00	00	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	1.0	00	00	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	.11	00	00	00	.89	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

LPP (SVM)

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	.89	00	00	.11	00	00	00	00	00	00
jack	00	1.0	00	00	00	00	00	00	00	00
jump	00	00	.78	00	00	00	.22	00	00	00
pjump	00	00	00	1.0	00	00	00	00	00	00
run	00	00	00	00	.78	00	.22	00	00	00
side	00	00	00	00	00	1.0	00	00	00	00
skip	00	00	.11	00	00	00	.89	00	00	00
walk	00	00	00	00	00	00	00	1.0	00	00
wave1	00	00	00	00	00	00	00	00	1.0	00
wave2	00	00	00	00	00	00	00	00	00	1.0

(b) SVM

Figure 4: Confusion matrices of different subspace learning methods

We also compare our result to state-of-the-art methods in Table 2. These methods are all tested on the same dataset [24]. Jia et al. [26] adopt a manifold embedding method, called Local Spatio-Temporal Discriminant Embedding (ASTDE), on human action recognition based on silhouette image frames of human action sequences. Kellokumpu et al. [27] describe human activities with texture features, i.e. extracting enhanced local binary pattern (LBP) histograms from Motion History Images and Motion Energy Images for action modelling. Wang and Suter [28] employ LPP to form low dimensional representation for silhouette based action recognition. In [29], Zhen *et al.* apply semi-supervised diffusion maps (SSDM) for dimensionality reduction and learning the eigen-space of the action silhouettes.

Table 2: Comparison with state-of-the-art.

Kellokumpu et al. (2008) [27]	97.8%
Jia et al. (2008) [26]	90.9%
Wang and Suter (2007) [28]	97.8%
Zhen *et al.* (2010) [29]	98.8%
Our methods	100%

4.3 Robustness test

We also test the robustness of our method on 10 irregularities of real-world actions, including walking when swinging the bag (1), walking with a briefcase (2), knee-up walking (3), limping (4), moon walking (5), no-feet (6), norm walking (7), pole (8), walking in skirt (9), walking with a dog (10). Figure 5 depicts the silhouettes of some frames in these sequences. The result of the robustness test is shown in Table 3, which shows that, except for four actions for which shape information is seriously changed by the irregular factors, our methods can handle irregularities quite well.

| (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |

Figure 5: Examples of the robustness test sequences.

Table 3: Result of the robustness test.

	Swing -bag	With briefcase	Knee- up	Limping	Moon -walk
SR-LDA	walk	walk	*jump*	walk	*run*
SR-KDA	walk	walk	*jump*	walk	*run*
	No feet	Norm -walk	*With -pole*	In skirt	*With dog*
SR-LDA	side	walk	*walk*	walk	*walk*
SR-KDA	side	walk	*side*	walk	*walk*

5. Conclusion

In this paper, a novel silhouette-based human action recognition algorithm using Histogram of Body Poses for action representation and Spectral Regression Discriminant Analysis for dimensionality reduction is proposed. The HBP descriptor captures the probabilistic distribution of body postures and is invariant to the length and frequency of the performed action. SR-LDA and SR-KDA are proved to be superior to other subspace learning methods, such as PCA, LPP and NPE, for preserving the intrinsic structure of action silhouette data. The combination of HBP and SRDA achieves 100% accuracy on the popular Weizmann action dataset, which is the best performance ever reported.

References

[1] D. Cai, X. He, and J. Han. Regularized locality preserving indexing via spectral regression. In ACM International Conference on Information and Knowledge Management, November 2007.

[2] D. Cai, X. He, and J. Han. Spectral regression for efficient regularized subspace learning. In IEEE International Conference on Computer Vision (ICCV), October 2007.

[3] M. Marszalek, I. Laptev and C. Schmid, Actions in Context, IEEE Conference on Computer Vision & Pattern Recognition (CVPR), Miami, USA, June 2009.

[4] M. S. Ryoo and J. K Aggarwal, Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities, International Conference on Computer Vision (ICCV), Kyoto, Japan, October 2009.

[5] L. Shao and R. Mattivi, Feature Detector and Descriptor Evaluation in Human Action Recognition, ACM International Conference on Image and Video Retrieval (CIVR), Xi'an, China, July 2010.

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," J. Cogn. Neurosci., vol. 3, no. 1, pp. 71–86, 1991.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2nd ed. New York: Wiley, 2000.

[8] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(3):328–340, Mar. 2005.

[9] J. B. Tenenbaum, V. de Silva, J. C. Langford,. A global geometric framework for nonlinear dimensionality reduction. Science, pages 2319{2323, 2000.

[10] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, pages 2323{2326, 2000.

[11] M. Belkin, and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, pages 1373{1396, 2003.

[12] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker. Geometric diffusions as a tool for harmonics amalysis and structure de ̄nition of data. Proc. Nat'l academy of sciences, 102(21):7426{7431, 2005.

[13] B. Sch ̈olkopf and A. J. Smola. Learning with Kernels. MIT Press, 2002.

[14] L. Wang, D. Suter, Recognizing Human Activities from Silhouettes: Motion Subspace and Factorial Discriminative Graphical Model, In Computer Vision and Pattern Recognition 2007, 1-8.

[15] S. Mika, G. R ̈atsch, J. Weston, B. Sch ̈olkopf, and K.-R. M ̈uller. Fisher discriminant analysis with kernels. In Proc. of IEEE Neural Networks for Signal Processing Workshop (NNSP), 1999.

[16] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. Neural Computation,(12):2385–2404, 2000.

[17] D. Cai, X. He, and J. Han. SRDA: An efficient algorithm for large scale discriminant analysis. IEEE Transactions on Knowledge and Data Engineering, 20(1):1–12, Jan. 2008.

[18] S. Guattery and G. L. Miller. Graph embeddings and laplacian eigenvalues. SIAM Journal on Matrix Analysis and Applications,21(3):703–723, 2000.

[19] F. R. K. Chung. Spectral Graph Theory, volume 92 of Regional Conference Series in Mathematics. AMS, 1997.

[20] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.

[21] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Advances in Neural Information Processing Systems 14, pages 585–591.

[22] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500):2323–2326, 2000.

[23] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer-Verlag, 2001.

[24] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as Space-Time Shapes. IEEE International Conference on Computer Vision, 17-21 Oct. 2005, vol. 2, 1395 –1402.

[25] C. C. Chang, C. J. Lin. LIBSVM: a library for support vector machines, 2001.

[26] K. Jia and D. Yeung, Human action recognition using Local Spatio-Temporal Discriminant Embedding, IEEE

Conference on Computer Vision and Pattern Recognition, 2008, Pages: 1-8.

[27] V. Kellokumpu, G. Zhao, M. Pietikänen. Texture based description of movements for activity analysis. Proc. Third International Conference on Computer Vision Theory and Applications (Madeira, Portugal, 2008), Vol.1, 206-213.

[28] L. Wang, D. Suter, Learning and Matching of Dynamic Shape Manifolds for Human Action Recognition, IEEE Transactions on Image Processing, June 2007, Vol.16, Issue:6, 1646-1661.

[29] F. Zheng, L. Shao and Z. Song, Eigen-space Learning Using Semi-supervised Diffusion Maps for Human Action Recognition, ACM International Conference on Image and Video Retrieval (CIVR), Xi'an, China, July 2010.